Resolução livro

Grupo: Eduardo Garcez, Davi Augusto, Gabriel Netto, João Arend e Luan Frederico.

Contents

Contexto

Calibração de doses de radiação - Regressão Poisson;

Dado problema apresentado queremos partir dos dados apresentados nas tabelas 9.9 e 9.10 e reproduzi-las com auxílio do OPENBUGS. A ideia é termos resultados aproximados aos das tabelas 9.11 e 9.12;

Pré-requisitos

```r
# install.packages("R2OpenBUGS")
library(R2OpenBUGS)
# install.packages("knitr")
library(knitr)
# install.packages("kableExtra")
library(kableExtra)
# install.packages("dplyr")
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following object is masked from 'package:kableExtra':
##
##     group_rows
## The following objects are masked from 'package:stats':
##
##     filter, lag
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

Dados 9.10

Tabela 9.10:

Tabela 9.10: Dados de radiação – *The Lundsteen-Piper (1989) lymphocyte data.*

i	Nº de aberrações y_i	Nº de células (1000) n_i	nível de dose d_i(rad)
1	0	585	0.10
2	3	1002	0.20
3	5	472	0.50
4	14	493	1.00
5	30	408	1.50
6	75	690	2.00
7	46	291	3.00
f_1	20	700	d_{f_1}

```r
# 1. Calibração (Tabela 9.10)
y_cal <- c(0, 3, 5, 14, 30, 75, 46)                 # Nº de aberrações
n_cal <- c(585, 1002, 472, 493, 408, 690, 291)      # Nº de células
d_cal <- c(0.10, 0.20, 0.50, 1.00, 1.50, 2.00, 3.00) # Nível de dose
m <- length(y_cal) # Número de observações de calibração

# 2. Indivíduo futuro (Linha 'f' da Tabela 9.10)
y_f <- 20   # Nº de aberrações observado
n_f <- 700 # Nº de células do futuro indivíduo

# Priori dos Betas ~ Normal(0, var=10000).
# BUGS usa PRECISÃO = 1/variância
prior_mean <- 0
prior_prec <- 1 / 10000

# 4. PREPARAR A LISTA DE DADOS PARA O OPENBUGS
bugs_data <- list(
  y = y_cal,
  n = n_cal,
  d = d_cal,
  m = m,
  yf = y_f,
```

```r
  nf = n_f,
  b_mean = prior_mean,
  b_prec = prior_prec
)

# print(bugs_data)
```

Modelo (i)

tentaremos replicar a primeira linha de resultados da Tabela 9.12: as estimativas para o Modelo(i).

Modelo (i): $Y \sim Poi(\mu)$ com $\log(\mu) = \beta_0 + \beta_1 \log(n) + \beta_2 \log(d)$.

Modelo OPENBUGS

Agora, criamos o arquivo de modelo. O texto o define como Modelo (i): $Y \sim Poi(\mu)$ com $\log(\mu) = \beta_0 + \beta_1 \log(n) + \beta_2 \log(d)$.

portanto teremos um *model1.txt* com o conteúdo:

```
# Modelo (i):
model {
# Verossimilhança ---
for (i in 1:m) {
    # Modelo Poisson
    y[i] ~ dpois(mu[i])

    # Modelo de Regressão Log-linear
    log(mu[i]) <- beta0 + beta1 * log(n[i]) + beta2 * log(d[i])
}

# Priori dos Parâmetros de Regressão (Vagas) ---
# Priori vaga conforme o texto (Normal com var=10000)
beta0 ~ dnorm(b_mean, b_prec)
beta1 ~ dnorm(b_mean, b_prec)
beta2 ~ dnorm(b_mean, b_prec)

# Indivíduo Futuro 'f' ---
yf ~ dpois(muf)
log(muf) <- beta0 + beta1 * log(nf) + beta2 * log(df) # df é desconhecido

# df ~ Ga(10, 0.1)
df ~ dgamma(df_a, df_b)
}
```

Execução

```r
# Chutes Iniciais ---

inits <- function() {
  list(
    beta0 = 0,
    beta1 = 1,
    beta2 = 1,
    df    = 1
```

```
  )
}

# Parâmetros para retornar ---
retorno <- c("beta0", "beta1", "beta2")

model_run_1 <- bugs(
  data = bugs_data,
  inits = inits,
  parameters.to.save = retorno,
  model.file = "model1.txt",
  n.chains = 3,
  n.iter = 10000,
  n.burnin = 5000,
  n.thin = 5
)

print(model_run_1, digits = 3)
```

```
## Inference for Bugs model at "model1.txt",
## Current: 3 chains, each with 10000 iterations (first 5000 discarded), n.thin = 5
## Cumulative: n.sims = 15000 iterations saved
##            mean    sd   2.5%    25%    50%    75%  97.5%  Rhat n.eff
## beta0    -5.190 1.400 -7.866 -6.173 -5.218 -4.219 -2.472 1.035    73
## beta1     1.282 0.219  0.854  1.131  1.287  1.435  1.701 1.028    92
## beta2     1.633 0.161  1.330  1.522  1.627  1.739  1.959 1.007   350
## deviance 33.608 2.377 30.860 31.860 33.000 34.710 39.820 1.007   350
##
## For each parameter, n.eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor (at convergence, Rhat=1).
##
## DIC info (using the rule, pD = Dbar-Dhat)
## pD = 2.939 and DIC = 36.550
## DIC is an estimate of expected predictive error (lower deviance is better).
```

Modelo (ii)

$Y \sim Poi(\mu)$ com $\mu = \alpha n d$

- Modelo (ii-a): Informação a priori vaga, com $a = b = 0$.
- Modelo (ii-b): Informação a priori informativa, com $a = 10$ e $b = 1000$.

```
# Usamos a aproximação MCMC padrão dgamma(0.001, 0.001)
bugs_data_2a <- bugs_data
bugs_data_2a$alpha_a <- 0.001
bugs_data_2a$alpha_b <- 0.001

# Modelo (ii-b) [Prior Informativa]
bugs_data_2b <- bugs_data
bugs_data_2b$alpha_a <- 10
bugs_data_2b$alpha_b <- 1000
```

Modelo OPENBUGS

```
# Modelo (ii):
model {
  # Verossimilhança ---
  for (i in 1:m) {
    y[i] ~ dpois(mu[i])
    mu[i] <- alpha * n[i] * d[i]
  }

  # Priori para Alpha ---
  # alpha ~ Ga(a, b)
  alpha ~ dgamma(alpha_a, alpha_b)

  # Indivíduo Futuro 'f'
  yf ~ dpois(muf)
  muf <- alpha * nf * df # df é desconhecido

  # Priori para df ~ Ga(10, 0.1)
  df ~ dgamma(df_a, df_b)
}
```

Execução

```r
# Chutes iniciais
inits_2 <- function() {
  list(
    alpha = 1,
    df    = 1
  )
}

# Parâmetros ---
retorno_2 <- c("alpha")

# Modelo (ii-a) [Prior Vaga] ---
model_run_2a <- bugs(
  data = bugs_data_2a,
  inits = inits_2,
  parameters.to.save = retorno_2,
  model.file = "model2.txt",
  n.chains = 3,
  n.iter = 10000,
  n.burnin = 5000,
  n.thin = 5
)

# Modelo (ii-b) [Prior Informativa] ---

model_run_2b <- bugs(
  data = bugs_data_2b,
  inits = inits_2,
  parameters.to.save = retorno_2,
  model.file = "model2.txt",
  n.chains = 3,
  n.iter = 10000,
```

```r
  n.burnin = 5000,
  n.thin = 5
)
```

Modelo (ii-a) [Prior Vaga]

```r
print(model_run_2a, digits = 6)
```

```
## Inference for Bugs model at "model2.txt",
## Current: 3 chains, each with 10000 iterations (first 5000 discarded), n.thin = 5
## Cumulative: n.sims = 15000 iterations saved
##                mean       sd     2.5%      25%      50%      75%    97.5%
## alpha     0.044927 0.003438  0.03853  0.04255  0.04483  0.04719  0.05194
## deviance 50.742777 1.408858 49.73000 49.83000 50.19000 51.08250 54.76000
##              Rhat n.eff
## alpha    1.000987 15000
## deviance 1.001115  9200
##
## For each parameter, n.eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor (at convergence, Rhat=1).
##
## DIC info (using the rule, pD = Dbar-Dhat)
## pD = 1.013000 and DIC = 51.760000
## DIC is an estimate of expected predictive error (lower deviance is better).
```

Modelo (ii-b) [Prior Informativa]

```r
print(model_run_2b, digits = 6)
```

```
## Inference for Bugs model at "model2.txt",
## Current: 3 chains, each with 10000 iterations (first 5000 discarded), n.thin = 5
## Cumulative: n.sims = 15000 iterations saved
##                mean       sd     2.5%      25%      50%      75%    97.5%
## alpha     0.037719 0.002790  0.03252  0.03581  0.03765  0.03956  0.04340
## deviance 55.640260 4.308766 49.94000 52.39000 54.79000 57.96000 65.95025
##              Rhat n.eff
## alpha    1.000904 15000
## deviance 1.000906 15000
##
## For each parameter, n.eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor (at convergence, Rhat=1).
##
## DIC info (using the rule, pD = Dbar-Dhat)
## pD = 0.947700 and DIC = 56.590000
## DIC is an estimate of expected predictive error (lower deviance is better).
```

Modelo (iii)

A definição do modelo é $Y \sim Poi(\mu)$ com $\mu = n(\alpha_0 + \alpha_1 d + \alpha_2 d^2)$

Vamos replicar:

- Modelo: (iii-a): $\alpha_0, \alpha_1, \alpha_2 \sim N(0, 10000)$.

- Modelo (iii-b): $\alpha_0, \alpha_1, \alpha_2 \sim Ga(1, 1)$.
- Modelo (iii-c): $\alpha_0, \alpha_1, \alpha_2 \sim Ga(1, 4)$.

```r
# Preparar dados para o Modelo (iii-a) [Prior N(0, 1000)]
bugs_data_3a <- bugs_data
bugs_data_3a$alpha_a <- 0
bugs_data_3a$alpha_b <- 1 / 10000

# Preparar dados para o Modelo (iii-b) [Prior Ga(1, 1)]
bugs_data_3b <- bugs_data
bugs_data_3b$alpha_a <- 1.0
bugs_data_3b$alpha_b <- 1.0

# Preparar dados para o Modelo (iii-c) [Prior Ga(1, 4)]
bugs_data_3c <- bugs_data
bugs_data_3c$alpha_a <- 1.0
bugs_data_3c$alpha_b <- 4.0
```

Modelo OPENBUGS Normal

```
# Modelo (iii-a):
model {
  for (i in 1:m) {
    y[i] ~ dpois(mu[i])
    mu[i] <- n[i] * (alpha0 + alpha1 * d[i] + alpha2 * pow(d[i], 2))
  }

  alpha0 ~ dnorm(b_mean, b_prec)
  alpha1 ~ dnorm(b_mean, b_prec)
  alpha2 ~ dnorm(b_mean, b_prec)
}
```

Modelo OPENBUGS Gamma

```
# Modelo (iii-b/c):
model {
  for (i in 1:m) {
    y[i] ~ dpois(mu[i])
    mu[i] <- n[i] * (alpha0 + alpha1 * d[i] + alpha2 * pow(d[i], 2))
  }

  alpha0 ~ dgamma(alpha_a, alpha_b)
  alpha1 ~ dgamma(alpha_a, alpha_b)
  alpha2 ~ dgamma(alpha_a, alpha_b)
}
```

Execução

```r
# Chutes iniciais
inits_3 <- function() {
  list(
    alpha0 = 1,
    alpha1 = 2,
    alpha2 = 3
  )
```

```r
}
retorno_3 <- c("alpha0", "alpha1", "alpha2")

model_run_3a <- bugs(
  data = bugs_data_3a,
  inits = inits_3,
  parameters.to.save = retorno_3,
  model.file = "model3_norm.txt",
  n.chains = 3,
  n.iter = 10000,
  n.burnin = 5000,
  n.thin = 5
)

# Executar o Modelo (iii-c) [Prior Ga(1, 1)] ---
model_run_3b <- bugs(
  data = bugs_data_3b,
  inits = inits_3,
  parameters.to.save = retorno_3,
  model.file = "model3_gamma.txt",
  n.chains = 3,
  n.iter = 10000,
  n.burnin = 5000,
  n.thin = 5
)

# Executar o Modelo (iii-c) [Prior Ga(1, 4)] ---

model_run_3c <- bugs(
  data = bugs_data_3c,
  inits = inits_3,
  parameters.to.save = retorno_3,
  model.file = "model3_gamma.txt",
  n.chains = 3,
  n.iter = 10000,
  n.burnin = 5000,
  n.thin = 5
)
```

Modelo (iii-a) [Prior N(0, 10000)]

```r
print(model_run_3a, digits = 7)
```

```
## Inference for Bugs model at "model3_norm.txt",
## Current: 3 chains, each with 10000 iterations (first 5000 discarded), n.thin = 5
## Cumulative: n.sims = 15000 iterations saved
##                  mean           sd      2.5%       25%       50%      75%
## alpha0      0.6913098    0.2912103    0.2987    0.3878    0.6527    1.053
## alpha1     -1.7233085    0.4464796   -2.3900   -2.3060   -1.5270   -1.332
## alpha2      1.1137502    0.1786009    0.8200    0.9264    1.1580    1.262
## deviance 6475.1941333 1524.9983732 4231.0000 4810.0000 6828.0000 7338.000
##                 97.5%      Rhat n.eff
## alpha0      1.099000 12.225040     3
```

```
## alpha1     -1.245000 25.113379     3
## alpha2      1.354025  5.513236     3
## deviance 9499.000000  5.073738     3
##
## For each parameter, n.eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor (at convergence, Rhat=1).
##
## DIC info (using the rule, pD = Dbar-Dhat)
## pD = 83.2400000 and DIC = 6558.0000000
## DIC is an estimate of expected predictive error (lower deviance is better).
```

Modelo (iii-b) [Prior Ga(1, 1)]

```r
print(model_run_3b, digits = 7)
```

```
## Inference for Bugs model at "model3_gamma.txt",
## Current: 3 chains, each with 10000 iterations (first 5000 discarded), n.thin = 5
## Cumulative: n.sims = 15000 iterations saved
##                mean        sd       2.5%        25%       50%       75%
## alpha0    0.0011387 0.0010396  0.0000336  0.0003631 8.441e-04  0.001609
## alpha1    0.0128891 0.0070324  0.0014140  0.0076067 1.225e-02  0.017420
## alpha2    0.0175137 0.0039616  0.0092378  0.0149000 1.773e-02  0.020330
## deviance 38.2658607 2.5142204 35.1000000 36.4374997 3.773e+01 39.470000
##              97.5%     Rhat n.eff
## alpha0    0.003845 1.001130  8600
## alpha1    0.028200 1.001012 15000
## alpha2    0.024630 1.000992 15000
## deviance 44.810250 1.001191  6800
##
## For each parameter, n.eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor (at convergence, Rhat=1).
##
## DIC info (using the rule, pD = Dbar-Dhat)
## pD = 1.8560000 and DIC = 40.1200000
## DIC is an estimate of expected predictive error (lower deviance is better).
```

Modelo (iii-c) [Prior Ga(1, 4)]

```r
print(model_run_3c, digits = 7)
```

```
## Inference for Bugs model at "model3_gamma.txt",
## Current: 3 chains, each with 10000 iterations (first 5000 discarded), n.thin = 5
## Cumulative: n.sims = 15000 iterations saved
##                mean        sd       2.5%       25%        50%        75%
## alpha0    0.0011450 0.0010423  0.0000318 3.676e-04  0.0008502  0.0016302
## alpha1    0.0129006 0.0069457  0.0014369 7.695e-03  0.0124100  0.0173500
## alpha2    0.0174716 0.0039201  0.0093269 1.486e-02  0.0176900  0.0202600
## deviance 38.2552880 2.4981978 35.0900000 3.644e+01 37.7049997 39.4700000
##              97.5%     Rhat n.eff
## alpha0    0.003889 1.000913 15000
## alpha1    0.028070 1.001040 14000
## alpha2    0.024550 1.001065 12000
## deviance 44.540000 1.000959 15000
##
```

```
## For each parameter, n.eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor (at convergence, Rhat=1).
##
## DIC info (using the rule, pD = Dbar-Dhat)
## pD = 1.8380000 and DIC = 40.0900000
## DIC is an estimate of expected predictive error (lower deviance is better).
```

Comparando as tabelas

```r
extract_summary <- function(model_run, model_name) {

  summary_df <- as.data.frame(model_run$summary)
  summary_df$Parametro <- rownames(model_run$summary)
  summary_df$Modelo <- model_name
  summary_df <- subset(summary_df, Parametro != "deviance")
  summary_df <- summary_df %>%
    select(
      Modelo,
      "Parâmetro" = Parametro,
      "média" = mean,
      "desvio padrão" = sd,
      "2.5%" = `2.5%`,
      Mediana = `50%`,
      "97.5%" = `97.5%`
    )

  return(summary_df)
}

tabela_m1 <- extract_summary(model_run_1,   "(i)")
tabela_m2a <- extract_summary(model_run_2a, "(ii-a)")
tabela_m2b <- extract_summary(model_run_2b, "(ii-b)")
tabela_m3a <- extract_summary(model_run_3a, "(iii-a)")
tabela_m3b <- extract_summary(model_run_3b, "(iii-b)")
tabela_m3c <- extract_summary(model_run_3c, "(iii-c)")

tabela_resumo_final <- bind_rows(
  tabela_m1,
  tabela_m2a,
  tabela_m2b,
  tabela_m3a,
  tabela_m3b,
  tabela_m3c
)

kable(tabela_resumo_final, booktabs = TRUE , row.names = FALSE) %>%
  kable_styling(latex_options = "hold_position") %>%
  collapse_rows(columns = 1, valign = "middle", latex_hline = "major")
```

Modelo	Parâmetro	média	desvio padrão	2.5%	Mediana	97.5%
(i)	beta0	-5.1899695	1.4003610	-7.8660000	-5.2180000	-2.471925
	beta1	1.2820438	0.2191942	0.8541825	1.2870000	1.701000
	beta2	1.6330109	0.1614093	1.3300000	1.6270000	1.959000

(ii-a)	alpha	0.0449271	0.0034382	0.0385297	0.0448300	0.051940
(ii-b)	alpha	0.0377191	0.0027903	0.0325200	0.0376500	0.043400
(iii-a)	alpha0	0.6913098	0.2912103	0.2987000	0.6527000	1.099000
	alpha1	-1.7233085	0.4464796	-2.3900000	-1.5270000	-1.245000
	alpha2	1.1137502	0.1786009	0.8200000	1.1580000	1.354025
(iii-b)	alpha0	0.0011387	0.0010396	0.0000336	0.0008441	0.003845
	alpha1	0.0128891	0.0070324	0.0014140	0.0122500	0.028200
	alpha2	0.0175137	0.0039616	0.0092378	0.0177300	0.024630
(iii-c)	alpha0	0.0011450	0.0010423	0.0000318	0.0008502	0.003889
	alpha1	0.0129006	0.0069457	0.0014369	0.0124100	0.028070
	alpha2	0.0174716	0.0039201	0.0093269	0.0176900	0.024550

Tabela 9.12:

Tabela 9.12: Estimativas pontuais e intervalos de credibilidade 95% para os parâmetros dos modelos – Dados da Tabela 9.10.*

Modelo	Parâmetro	média	desvio padrão	2.5%	mediana	97.5%
(i)	β_0	-5.196	1.441	-8.031	-5.184	-2.4
	β_1	1.282	0.2248	0.8425	1.281	1.725
	β_2	1.638	0.1665	1.328	1.635	1.973
(ii-a)	α	0.044901	0.003414	0.038210	0.044901	0.051592
(ii-b)	α	0.037709	0.002788	0.032246	0.037709	0.043173
(iii-a)	α_0	-0.004058	0.002652	-0.009358	-0.003989	1.11E-03
	α_1	0.03266	0.01276	0.008598	0.03252	0.05873
	α_2	0.009432	0.005776	-0.001981	0.009382	0.02064
(iii-b)	α_0	0.001128	0.001041	3.12E-05	8.33E-04	0.003824
	α_1	0.01309	0.00715	0.001317	0.01246	0.02882
	α_2	0.01742	0.004023	0.008996	0.01762	0.02458
(iii-c)	α_0	0.00114	0.001038	3.27E-05	8.43E-04	0.003831
	α_1	0.01292	0.007111	0.001256	0.01234	0.02825
	α_2	0.01749	0.004002	0.00922	0.01767	0.02469

Dados 9.9

Os princípios e modelos são exatamente os mesmos do anterior então faremos de forma mais direta:

Tabela 9.9:

Tabela 9.9: Dados de radiação por neutrões Po-Be.

i	Nº de aberrações y_i	Nº de células (1000) n_i	nível de dose d_i(rad)
1	109	269	50
2	47	78	75
3	94	115	100
4	114	90	150
5	138	84	200
6	125	59	250
7	97	37	300
f_1	64	104	d_{f_1}
f_2	8	13	d_{f_2}

```r
# Dados de calibração (i=1...7)
y_cal_T9 <- c(109, 47, 94, 114, 138, 125, 97)
n_cal_T9 <- c(269, 78, 115, 90, 84, 59, 37)
d_cal_T9 <- c(50, 75, 100, 150, 200, 250, 300)
m_T9 <- length(y_cal_T9)

# Priori vaga (Normal)
prior_mean <- 0.0
prior_prec <- 1.0 / 10000.0

# Lista de dados base (apenas m=7 dados)
bugs_data_T9_base <- list(
  y = y_cal_T9,
  n = n_cal_T9,
  d = d_cal_T9,
  m = m_T9,
  b_mean = prior_mean,
  b_prec = prior_prec
)

n_iter   <- 10000
n_burn   <- 5000
n_thin   <- 5
n_chains <- 3

lista_resumos_T9 <- list()

## Executar Modelos ---
```

```r
# Modelo (i)

params_1 <- c("beta0", "beta1", "beta2")
inits_1 <- function() list(beta0 = 0, beta1 = 1, beta2 = 1)

run_m1 <- bugs(
  data = bugs_data_T9_base,
  inits = inits_1,
  parameters.to.save = params_1,
  model.file = "model1.txt",
  n.chains = n_chains,
  n.iter = n_iter,
  n.burnin = n_burn,
  n.thin = n_thin
)

lista_resumos_T9$m1 <- extract_summary(run_m1, "(i)")

# Modelo (ii)
params_2 <- c("alpha")
inits_2 <- function() list(alpha = 0.01)
data_2a <- bugs_data_T9_base
data_2a$alpha_a <- 0.001
data_2a$alpha_b <- 0.001
run_m2a <- bugs(
  data = data_2a,
  inits = inits_2,
  parameters.to.save = params_2,
  model.file = "model2.txt",
  n.chains = n_chains,
  n.iter = n_iter,
  n.burnin = n_burn,
  n.thin = n_thin
)

lista_resumos_T9$m2a <- extract_summary(run_m2a, "(ii-a)")

# Modelo (ii-b)
data_2b         <- bugs_data_T9_base
data_2b$alpha_a <- 10.0
data_2b$alpha_b <- 1000.0
run_m2b <- bugs(
  data = data_2b,
  inits = inits_2,
  parameters.to.save = params_2,
  model.file = "model2.txt",
  n.chains = n_chains,
  n.iter = n_iter,
  n.burnin = n_burn,
  n.thin = n_thin
)

lista_resumos_T9$m2b <- extract_summary(run_m2b, "(ii-b)")
```

```r
# Modelo (iii-a)
params_3 <- c("alpha0", "alpha1", "alpha2")
inits_3 <- function() list(alpha0 = 0.01, alpha1 = 0.01, alpha2 = 0.00001)
run_m3a <- bugs(
  data = bugs_data_T9_base,
  inits = inits_3,
  parameters.to.save = params_3,
  model.file = "model3_norm.txt",
  n.chains = n_chains,
  n.iter = n_iter,
  n.burnin = n_burn,
  n.thin = n_thin
)
lista_resumos_T9$m3a <- extract_summary(run_m3a, "(iii-a)")

# Modelo (iii-b)

data_3b <- bugs_data_T9_base
data_3b$alpha_a <- 1.0
data_3b$alpha_b <- 1.0
run_m3b <- bugs(
  data = data_3b,
  inits = inits_3,
  parameters.to.save = params_3,
  model.file = "model3_gamma.txt",
  n.chains = n_chains,
  n.iter = n_iter,
  n.burnin = n_burn,
  n.thin = n_thin
)
lista_resumos_T9$m3b <- extract_summary(run_m3b, "(iii-b)")

# Modelo (iii-c)
data_3c <- bugs_data_T9_base
data_3c$alpha_a <- 1.0
data_3c$alpha_b <- 4.0
run_m3c <- bugs(
  data = data_3c,
  inits = inits_3,
  parameters.to.save = params_3,
  model.file = "model3_gamma.txt",
  n.chains = n_chains,
  n.iter = n_iter,
  n.burnin = n_burn,
  n.thin = n_thin
)

lista_resumos_T9$m3c <- extract_summary(run_m3c, "(iii-c)")

tabela_resumo_T9 <- bind_rows(lista_resumos_T9)

kable(tabela_resumo_T9, booktabs = TRUE , row.names = FALSE) %>%
  kable_styling(latex_options = "hold_position") %>%
```

```r
collapse_rows(columns = 1, valign = "middle", latex_hline = "major")
```

Modelo	Parâmetro	média	desvio padrão	2.5%	Mediana	97.5%
(i)	beta0	-5.0912398	1.2905787	-7.8170500	-5.0230000	-2.6019250
	beta1	1.0120338	0.1440128	0.7285875	1.0030000	1.3080000
	beta2	1.0496571	0.1365069	0.7892975	1.0450000	1.3320250
(ii-a)	alpha	0.0083255	0.0003096	0.0077290	0.0083190	0.0089380
(ii-b)	alpha	0.0083451	0.0003068	0.0077570	0.0083400	0.0089470
(iii-a)	alpha0	0.1088060	0.1090262	-0.1024025	0.1321000	0.2788075
	alpha1	0.0057588	0.0020775	0.0031260	0.0049010	0.0095450
	alpha2	0.0000100	0.0000073	-0.0000021	0.0000135	0.0000190
(iii-b)	alpha0	0.0984627	0.0719273	0.0043989	0.0850100	0.2685025
	alpha1	0.0061013	0.0014066	0.0028430	0.0063295	0.0081740
	alpha2	0.0000085	0.0000052	0.0000007	0.0000079	0.0000204
(iii-c)	alpha0	0.0845552	0.0642533	0.0035457	0.0711950	0.2412000
	alpha1	0.0063374	0.0012704	0.0032947	0.0065490	0.0081940
	alpha2	0.0000078	0.0000048	0.0000006	0.0000071	0.0000189

Tabela 9.11:

Tabela 9.11: Estimativas pontuais e intervalos de credibilidade 95% para os parâmetros dos modelos – Dados da Tabela 9.9.*

Modelo	Parâmetro	média	desvio padrão	2.5%	mediana	97.5%
(i)	β_0	-4.752	1.39	-7.702	-4.701	-2.095
	β_1	0.9763	0.1549	0.6876	0.9719	1.294
	β_2	1.014	0.1477	0.7343	1.009	1.324
(ii-a)	α	0.008327	0.000309	0.007720	0.008327	0.008933
(ii-b)	α	0.008346	0.000308	0.007742	0.008346	0.008949
(iii-a)	α_0	0.01092	0.1202	-0.2093	0.01343	0.2523
	α_1	0.007805	0.002366	0.003093	0.007676	0.01217
	α_2	2.80E-06	8.29E-06	-1.25E-05	3.15E-06	1.91E-05
(iii-b)	α_0	0.09803	0.07159	0.005422	0.08338	0.2732
	α_1	0.006112	0.001407	0.002699	0.006365	0.008138
	α_2	8.43E-06	5.22E-06	6.54E-07	7.69E-06	2.06E-05
(iii-c)	α_0	0.08225	0.06138	0.003541	0.07004	0.228
	α_1	0.006361	0.001225	0.003612	0.006523	0.008224
	α_2	7.72E-06	4.67E-06	5.91E-07	7.17E-06	1.79E-05

Com exceção do modelo (iii-a) com os dados da tabela 9.10, podemos considerar que houve sucesso na reprodução das tabelas apresentadas pelo livro.